A MAXIMUM LIKELIHOOD TRACKER

Robert D. Conrad

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A MAXIMUM LIKELIHOOD
TRACKER

by

Robert D. Conrad

March 1981

Thesis Advisor: A. Washburn

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Maximum Likelihood Tracker		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; March 1981
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert D. Conrad		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1981
		13. NUMBER OF PAGES 60
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Tracking, Kalman filter, False alarms

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report deals with the development and utilization of a maximum likelihood tracking algorithm designed to handle a single diffusing target. The tracker is required to accept or reject each of a sequence

of discrete position reports, some of which are false alarms. Average tracking time, E(T), is defined and used to evaluate tracker performance. The effects of tracker memory are examined and a Kalman filter is developed to handle the effects of measurement error. Simulation results are included.

Approved for public release; distribution unlimited.

A Maximum Likelihood Tracker

by

Robert D. Conrad
Lieutenant, United States Navy
B.S., United States Naval Academy, 1975

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1981

ABSTRACT

This report deals with the development and utilization of a maximum likelihood tracking algorithm designed to handle a single diffusing target. The tracker is required to accept or reject each of a sequence of discrete position reports, some of which are false alarms. Average tracking time, $E(T)$, is defined and used to evaluate tracker performance. The effects of tracker memory are examined and a Kalman filter is developed to handle the effects of measurement error. Simulation results are included.

TABLE OF CONTENTS

I. INTRODUCTION

The treatment of false alarms is central to the development of any useful set of tracking decision rules. The purpose of this report is to examine the effectiveness of a specific tracking algorithm to be used in the presence of false alarms. A maximum likelihood decision rule will be developed and used in conjunction with an expandable tracker memory. The primary measure of tracker effectiveness will be average tracking time, E(T).

A. BACKGROUND

Every tracking system must in some way account for the possibility of receiving both real and false target position reports. Tracking radars typically employ a 'gate' criterion that rejects position reports which differ significantly from what target dynamics might permit. The gate helps to maintain track in a relatively noisy environment at the expense of occasionally rejecting a real detection. Surveillance systems with data rates considerably below those of tracking radars, such as might be found in an ASW environment, must also deal with false detections, and these will be the subject of this report.

Tracking systems with low data rates must necessarily take a different approach than a typical aircraft tracking system might employ. Most rapid rate tracking systems are judged by their ability to recognize and differentiate new tracks, crossing tracks, split tracks,

and fading tracks. They must be able to process large amounts of information, quickly and efficiently, to produce precise target position estimates on a real time basis. In his article 'An Optimal Data Association Problem in Surveillance Theory' Robert Sittler developed a maximum likelihood model for just this purpose.

In a situation where the data rate is low, the emphasis shifts from how well we track to whether we can track at all. We will look at a particular class of trackers that maintains an estimate of target position at all times, and for which it is therefore possible to define tracking error at all times.

Figure 1 illustrates the results of three separate tracking sequences using the maximum likelihood memoryless tracker developed in this report. In the figure, error is measured in units of length and used to represent the difference between the target's real position and the tracker's estimated position of the target at any particular time. Due to the discrete nature of the target position reports used in the simulation, error is measured accurately only at the positions marked with a •. In all cases the error initially fluctuates up and down, but finally grows very rapidly. We wish to identify a "tracking time" that marks the end of the fluctuation region, and to define a tracking scheme that makes the average "tracking time" as large as possible. As figure 1 shows, a precise definition of tracking time can be illusive, and we will therefore define a conservative measure later in this chapter.

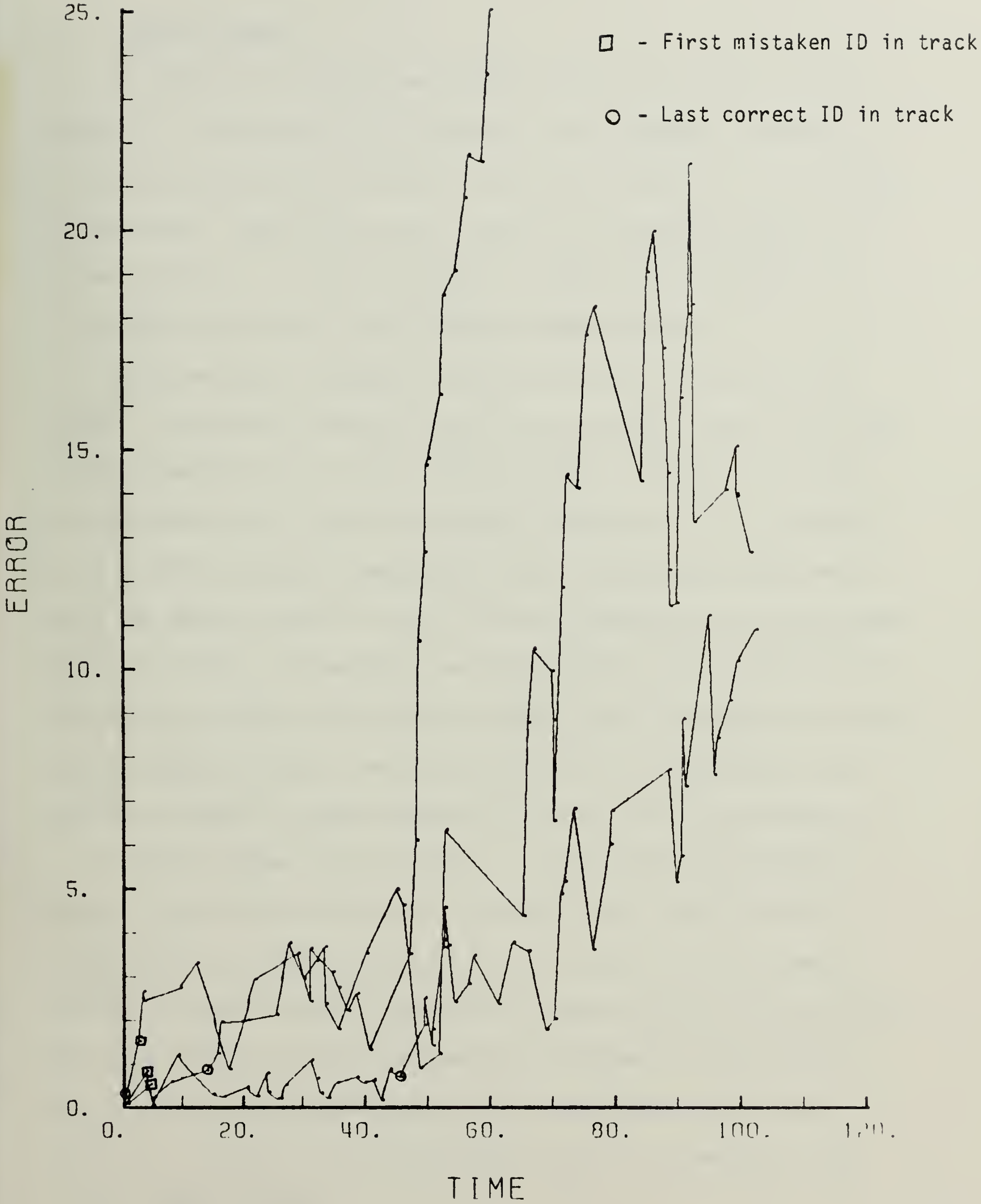

Figure 1. Time vs error plot for 3 sample runs of the maximum likelihood memoryless tracker.

B. PROBLEM DYNAMICS

We consider an abstract two-dimensional version of the tracking problem. Real target position reports arrive randomly, according to a Poisson process with data rate λ [units are (time^{-1})]. In a process of this kind, reports are equally likely to arrive at any point in a fixed interval. It is not possible to identify real detections by their time of arrival or the interval between arrivals.

Target motion is assumed to be due to normal diffusion with no drift. A diffusion constant D [units are $(\text{length}^2 \times \text{time}^{-1})$] will be used to characterize a single target's motion. Motion of this type can be modeled with a bivariate normal distribution. In a situation with no false alarms or measurement error, the target density would have the physical shape of a tall mountain centered on the last target position report. The mountain spreads gradually in all directions as the interval between detections increases, much like a mountain of snow melting under the sun. The motion of Fleet Ballistic Missile Submarines on patrol is often modeled with this type of distribution.

Tracking systems are often prone to some inherent measurement error. Many systems will have an internal bias in one direction, around which a detection might be likely to occur. In this report we will consider the case in which that type of bias can be recognized and accounted for. We will assume that the reported target position has a circular normal measurement error with standard deviation σ [units are (length) in each direction], and will be centered on the actual target position.

In most tracking systems there exists the possibility of a detection occurring in a location where no target exists. In a radar system it may be due to a random electrical disturbance, or with sonar, due to a transient school of fish. In this model, any detection not associated with the single real target will be called a false alarm. Every position report will therefore require classification as either a target contact or a false alarm. The case where other 'trackable' objects are present will not be considered.

False alarms will be considered to occur according to a space-time Poisson process with rate η [units are (length^2 x time^{-1})]. This results in false alarms occurring uniformly over an area with time spacing of events being governed by a Poisson process. This type of detection process can be assumed to arise from random disturbances to the system or from real but rare detections on other objects.

The assumption will also be made that there is no discernable difference between false and real detections. This assumption can be considered highly pessimistic when compared to any existing tracking system. The result is to drive the average tracking time toward a lower bound.

We have used four parameters, D, λ, η, and σ, to characterize the dynamics of the tracking problem. Since there are only two dimensions, time and length, actually involved, we can summarize the problem with two dimensionless summary measures. These are:

$$\alpha = \lambda/\sqrt{2\pi\eta D} = \text{rate ratio}$$

$$\beta = \sigma/\sqrt{D/\lambda} = \text{accuracy ratio}$$

C. TRACKER MEMORY

The concept of memory raises several possibilities when applied to tracking systems in general. Qualitatively, an expanded memory would seem desirable, but must be assumed to incur an additional cost to the tracker. Two questions addressed in this report are how much memory is needed and in what way should memory be used.

First we address the simplest tracking problem in which a single decision is made following each detection. Tracking begins with a known real target detection. Each subsequent position report is evaluated as real or false instantaneously. If evaluated false, the detection will be forgotten and the problem will continue. If evaluated real, the problem will be updated and restarted from the target's new position estimate. Tracking time is considered to stop at the time of the tracker's first mistake; i.e.

1) Tracker accepts a false alarm as real.

2) Tracker rejects a real detection.

These rules apply to what will be called the memoryless tracker.

Figure 1 shows the results of three separate tracking sequences simulated with the maximum likelihood memoryless model developed in this report. The time of the tracker's first mistaken identification is marked with a □ in each case. Also marked with a ○ is the last time, within the allotted simulation time, that the tracker correctly identified the real target. From the sample cases shown it is not clear that the tracker has actually lost his tracking ability after

making only one mistake. The tracker demonstrates an apparent ability to reacquire the target in some cases. It is clear though, that there is a point in time beyond which error will generally continue to increase. Any reduction in tracker error beyond this point can be considered to be almost exclusively due to the random chance that the target might walk back into the tracker rather than due to tracker efficiency. A problem arises however because of the fact that there is some positive probability that at some time in the future, the tracker might accidentally reacquire the target. A trivial tracker could, in fact, extend the tracking time indefinitely by accepting all detections that occur and defining tracking time to end at the time of the last correct identification. For this reason, we will use E(T), the average time till the tracker makes his first mistake, as our summary measure of tracker effectiveness and consider it a useful lower bound on actual tracking time.

The tracker with memory will face the same task and its performance will be judged in a similar manner. We look first at the case where the accuracy ratio, β, equals zero. In the diffusing target scenario, future target positions can then be considered to be independent of all but the real target's present position. Reports received prior to the most recent target detection will, therefore, not be retained in memory. Rather than demand an immediate decision on subsequent detections, however, the tracker with memory will be given the opportunity to file a finite number of detections in memory for a future decision.

Following each detection he will decide whether any single subset of detections on file constitutes the actual target track. If the decision is yes, memory will be scrubbed of all but the most recent position estimate and the process will begin again. When no track has been accepted, the tracker will retain all detections on file until his memory capacity is exceeded. As additional reports are received he will sequentially scrub detections on file according to a predetermined rule to be discussed later.

In the situation where the accuracy ratio, β, is not equal to zero, the problem is handled in essentially the same way. Successive real target detections are not independent of previous ones however, and will require special handling. This case will be dealt with in chapter 4.

So that effectiveness of the memoryless tracker and the tracker with memory may be compared, tracking time will be defined in essentially the same way as before. The tracker with memory will begin at time zero with a known real target detection and end when he makes his first mistake. Mistakes occur in any of the following ways:

1) A false alarm is included in an accepted set of detections.

2) A real detection is omitted from an accepted set of detections.

Tracking time is stopped at the time associated with the mistaken detection, which may be earlier than the time at which the fatal decision is made.

In the next chapter, we will develop the theory and specific decision rules for what will be called the maximum likelihood tracker. To simplify the analytic solution, measurement error will not be considered in that chapter or the next. In chapter 3 we will examine the results of a maximum likelihood tracker simulation and compare the results with those of Washburn [3]. A memory capacity will be given to the tracker in this chapter and its effect will be analyzed. In chapter 4 we will introduce and deal with measurement error. Chapter 5 will focus on the conclusions of this report.

II. MAXIMUM LIKELIHOOD DECISION RULE

In classical statistics the maximum likelihood estimator is typically used to estimate the value of a distributional parameter that has maximum chance of producing a sample result. In the study of a time-sequenced series of events the same concept can be applied. The subject of this chapter will be to develop a simplified decision rule that reflects the concept of maximum likelihood. The motivation lies behind the idea of labeling any series of detections in such a way as to maximize the probability of that set of labels being correct. The subset of detections labeled real in a series of detections could also be called the target track. The final result will be a simple summary measure by which every series of detection labels can be compared. Only the most likely series of events will be accepted.

A. SUMMARY OF TRACKING RULES

The problem of evaluating a single detection is particularly adaptable to a maximum likelihood model. We need only compare the probability of the detection being real against the probability of its being false. Given the characteristics of the diffusing target, this problem will yield easily to analysis.

When a series of K detections must be considered, the problem escalates in difficulty but follows the same basic principles. All possible combinations of real and false detection labels must be

compared. The combination of labels that holds the greatest chance of being correct would be accepted as real. The label combinations would be bounded by the set where all detections were labeled false as well as by the set where all were labeled real. All combinations within those bounds would have to be examined.

With this type of tracker it would be theoretically possible but practically unmanageable to retain all detections on file for future re-evaluation, even after they may have already been accepted. For the purposes of this report, the tracker will not be permitted to reconsider past decisions. The tracker can only progress from the last accepted decision.

The magnitude of this problem increases dramatically as additional detections are added to tracker memory. At each step the number of possible combinations which must be considered actually doubles. Accordingly, the size of tracker memory quickly becomes an issue of critical concern.

To simplify the analysis somewhat, the tracker's circular normal measurement error will not be considered in this chapter or the next. The inclusion of measurement error serves to complicate computer implementation of the decision process, although the maximum likelihood rules remain essentially unchanged. The case where measurement error is not zero will be dealt with in chapter 4.

To summarize the tracking rule that will be developed in the remainder of this chapter, we will first define a new concept for distance. Any two detections will have a dimensionless "distance"

between them equal to $[R^2/(2Dt)] + \ln(2\pi\eta\, Dt/\lambda)$ where R is the physical distance and t is the time difference between detections. A "track" then, will simply be a set of detections connected in order of occurrence, and the most likely track is the one with the shortest total distance. Note that the distance can be negative so that the best track is not necessarily null. In fact, the null solution corresponds to case where every detection is evaluated as being false. The rule presented here is actually a special case of the tracking scheme developed by Sittler [2].

B. DEVELOPMENT OF TRACKING RULES

To begin the analysis we consider a discrete approximation to the problem. We will assume time is discrete with intervals of length Δ. Space will be considered to be made of discrete sections with area of size δ. Figure 2 illustrates a possible sequence of two events in space and time. Although figure 2 shows only one dimension of space, x, a second dimension, y, will also be considered in the mathematics that follows. With two detections on file we must consider labeling the events in one of four possible ways

1) Detection one is real, two is false
2) Detection one is false, two is real
3) Detection one is real, two is also real
4) Detection one is false, two is also false

The likelihood of any of these label patterns or tracks can be expressed with a combination of the following expressions:

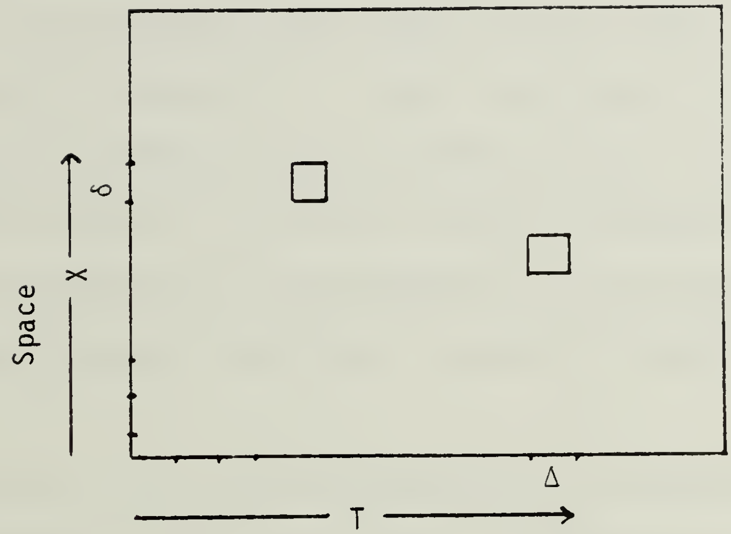

Figure 2. Sample space-time plot for two detections.

P[Real detection occurs at (X_i,Y_i) at time t_i] =

$$(\delta/(2\pi D t_i)\exp[-(X_i^2+Y_i^2)/(2Dt_i)])(\lambda\Delta) \qquad (1)$$

P[False alarm occurs at (X_i,Y_i) at time t_i] =

$$(\delta/A)(\eta A\Delta) = \delta\eta\Delta \qquad (2)$$

The first factor in expression (1) is simply the normal density function for a diffusing target multiplied by the elementary area δ. It represents the probability that the target is within a given elementary area at a specific time. The variable t_i is actually $n_i\Delta$, where n_i is

the number of time intervals, Δ, since the last detection labeled real in the series. N will be the total number of time intervals, Δ, included in the set. The word 'at' is being used loosely to indicate a detection within an elementary space and time interval. X_i and Y_i are the rectangular distances from the last detection considered real in the series. The second term in the expression is the Poisson probability of a real detection occurring in a specific elementary time interval Δ.

Expression (2) is the false alarm equivalent to expression (1). To avoid the issue of boundary conditions, in expression (1) we must define the area, A, in expression (2), to be large enough to effectively negate any possibility of the target reaching a boundary before either the tracker loses track or is able to reinitialize the problem.

Referring back to the four possible label combinations, the probabilities for each of these events can be expressed as follows:

$$P_1 = P[\text{R.D. in } \delta_1 \text{ when } n_1 \cdot \Delta = t_1] \cdot P[\text{F.D. in } \delta_2 \text{ when } n_2 \cdot \Delta = t_2]$$
$$\cdot\, P[\text{No R.D. in } (N-1)\Delta\text{'s}] \cdot P[\text{No F.D. in } (N-1)\Delta\text{'s}] \tag{3}$$

$$P_2 = P[\text{F.D. in } \delta_1 \text{ when } n_1 \cdot \Delta = t_1] \cdot P[\text{R.D. in } \delta_2 \text{ when } n_2 \cdot \Delta = t_2]$$
$$\cdot\, P[\text{No R.D. in } (N-1)\Delta\text{'s}] \cdot P[\text{No F.D. in } (N-1)\Delta\text{'s}] \tag{4}$$

$$P_3 = P[\text{R.D. in } \delta_1 \text{ when } n_1 \cdot \Delta = t_1] \cdot P[\text{R.D. in } \delta_2 \text{ when } n_2 \cdot \Delta = t_2]$$
$$\cdot\, P[\text{No R.D. in } (N-2)\Delta\text{'s}] \cdot P[\text{No F.D. in } (N)\ \Delta\text{'s}] \tag{5}$$

$$P_4 = P[\text{F.D. in } \delta_1 \text{ when } n_1 \cdot \Delta = t_1] \cdot P[\text{F.D. in } \delta_2 \text{ when } n_2 \cdot \Delta = t_2]$$

$$\cdot\ P[\text{No R.D. in N } \Delta\text{'s}] \cdot P[\text{No F.D. in } (N-2)\Delta\text{'s}] \qquad (6)$$

Then expressing (3) and (5) analytically we have:

$$P_1 = [(\lambda\Delta)\cdot\delta/(2\pi Dt_i)\exp[-(X_1{}^2+Y_1{}^2)/(2Dt_1)]]$$

$$\cdot\ [(\eta A\Delta)(\delta/A)][1-\lambda\Delta]^{T/\Delta-1}\ [1-\eta A\Delta]^{T/\Delta-1} \qquad (7)$$

$$P_3 = [(\lambda\Delta)\cdot\delta/(2\pi Dt_1)\exp[-(X_1{}^2+Y_1{}^2)/(2Dt_1)]]$$

$$\cdot\ [(\lambda\Delta)(\delta/(2\pi Dt_2)\exp[-(X_2{}^2+Y_2{}^2)/(2Dt_2)]]$$

$$\cdot\ [1-\lambda\Delta]^{T/\Delta-2}\ [1-\eta A\Delta]^{T/\Delta} \qquad (8)$$

where $T = N\,\Delta$ is the total time being considered. The remaining terms can be expressed in similar fashion.

The task now is to find a way in which to compare the various analytic probabilities. It can be seen immediately that each of the expressions will contain a number of equivalent factors.

Shifting momentarily to the general case, we will consider the situation where there are K detections in the set, m_a of which are being evaluated as real in track (a) and m_b of which are being evaluated as real in track (b). The likelihood ratio of the two tracks is

$$\frac{P_a}{P_b} = \frac{[(\lambda\Delta)^{m_a} \prod_{i=1}^{m_a} \delta/(2\pi D t_{ai}) \exp[-(X_{ai}^2+Y_{ai}^2)/(2Dt_{ai})]}{[(\lambda\Delta)^{m_b} \prod_{j=1}^{m_b} \delta/(2\pi D t_{bj}) \exp[-(X_{bj}^2+Y_{bj}^2)/(2Dt_{bj})]}$$

$$\cdot \frac{[(\delta/A)^{K-m_a} (\eta A\Delta)^{K-m_a}] (1-\lambda\Delta)^{T/\Delta-m_a} (1-\eta A\Delta)^{T/\Delta-(K-m_a)}}{[(\delta/A)^{K-m_b} (\eta A\Delta)^{K-m_b}] (1-\lambda\Delta)^{T/\Delta-m_b} (1-\eta A\Delta)^{T/\Delta-(K-m_b)}} \qquad (9)$$

which reduces to

$$\frac{P_a}{P_b} = \frac{[(\lambda\Delta)^{m_a} \prod_{i=1}^{m_a} \delta/(2\pi D t_{ai}) \exp[-(X_{ai}^2+Y_{ai}^2)/(2Dt_{ai})](\delta\eta\Delta)^{K-m_a}}{[(\lambda\Delta)^{m_b} \prod_{j=1}^{m_b} \delta/(2\pi D t_{bj}) \exp[-(X_{bj}^2+Y_{bj}^2)/(2Dt_{bj})](\delta\eta\Delta)^{K-m_b}}$$

$$\cdot (1-\lambda\Delta)^{m_b-m_a} (1-\eta A\Delta)^{m_a-m_b} \qquad (10)$$

The subscripts on X, Y, and t have been changed to indicate that there are different values associated with the terms from different paths; X_{ai} is the X-coordinate of the iTH detection on track (a), etc. Note that when Δ is small the last two factors in equation (10) are approximately equal to the value one. We will make this approximation, and by further cancellation we obtain

$$\frac{P_a}{P_b} = \frac{(\lambda/\eta)^{m_a} \prod_{i=1}^{m_a} 1/(2\pi D t_{ai}) \exp[-(X_{ai}^2+Y_{ai}^2)/(2Dt_{ai})]}{(\lambda/\eta)^{m_b} \prod_{j=1}^{m_b} 1/(2\pi D t_{bj}) \exp[-(X_{bj}^2+Y_{bj}^2)/(2Dt_{bj})]} \tag{11}$$

which suggests a useful decision rule. Note that the quantities A, Δ, and σ do not appear. Each term in the expression is either known or can be measured. A decision rule would be to pick track (a) if the ratio was greater than one, or pick track (b) if the ratio was less than one. The comparison could be carried out for all possible tracks in the set of detections with the winner being declared most likely. The method is intuitively reasonable and easily implemented by computer as long as the maximum number of possible tracks remains small.

For simplification, it is convenient to use the logarithmic form of equation (11). Let Li=ln Pi. Then Pi>Pj if and only if Li<Lj. The rule then would be to pick the track with the smallest value Li. The final summary measure for track (a) will be

$$L_a = \sum_{i=1}^{m_a} [(X_{ai}^2+Y_{ai}^2)/(2Dt_{ai}) + \ln(2\pi\eta D t_{ai}/\lambda)] \tag{12}$$

The criterion for track acceptance will be, accept the track with the smallest value Li. The case where all detections are labeled false will have the value zero. In this way we are actually choosing to

label the observed series of events with the pattern of detection types that is most likely to have occurred.

Note that the first term in equation 12 is always positive. The second term will be negative only when the ratio $(2\pi\eta DTi/\lambda)$ is less than one. The implication is that there is a limit in time beyond which a single detection can not be accepted. With no memory, the tracker will stall if he has not accepted a detection by a given time limit. With a memory capacity, the tracker is able to weight the various combinations of time and space differences between detections. In this way he may still accumulate an acceptable track.

As an example we can look back at the situation developed from figure 2. The four summary measures would be

$$1)\quad L_1 = (X_{11}^2+Y_{11}^2)/(2Dt_{11}) + \ln(2\pi\eta Dt_{11}/\lambda) \qquad (13)$$

$$2)\quad L_2 = (X_{21}^2+Y_{21}^2)/(2Dt_{21}) + \ln(2\pi\eta Dt_{21}/\lambda) \qquad (14)$$

$$3)\quad L_3 = (X_{31}^2+Y_{31}^2)/(2Dt_{31}) + \ln(X_{32}^2+Y_{32}^2)/(2Dt_{32}) + \ln(2\pi\eta Dt_{31}/\lambda) + \ln(2\pi\eta Dt_{32}/\lambda) \qquad (15)$$

$$4)\quad L_4 = 0 \qquad (16)$$

The trackers rule would be to accept the track with the smallest value, update the memory file accordingly, then wait for the next detection to occur.

III. TRACKING WITH NO MEASUREMENT ERROR

For the purpose of examining the value of a new tracking model, it is useful to make comparisons with a model already in existence. In this section we will examine results of simulating the maximum likelihood tracker and compare the results with those of Washburn [3], who has developed an optimized tracking rule for the memoryless tracker with no measurement error. We will then introduce memory to the maximum likelihood model and note the improved results.

A. THE MEMORYLESS TRACKER

Examining the same problem presented here, Washburn [3] arrives at an analytic solution which produces a tracking rule based on detection range as a function of time. Given the dynamic parameters of the problem and a detection time, his rule will generate a range from the last target position inside which the new detection should be accepted as real. Should the detection fall outside that range circle it would be rejected as false. The method is relatively easy to implement requiring only a single function evaluation for each detection that occurs. Based on this algorithm he is able to identify a dimensionless curve for λT, (detection rate x average tracking time), as a function of the rate ratio α. This curve is reproduced in figure 3. He has shown that these are maximum expected tracking times for the memoryless tracker with no measurement error.

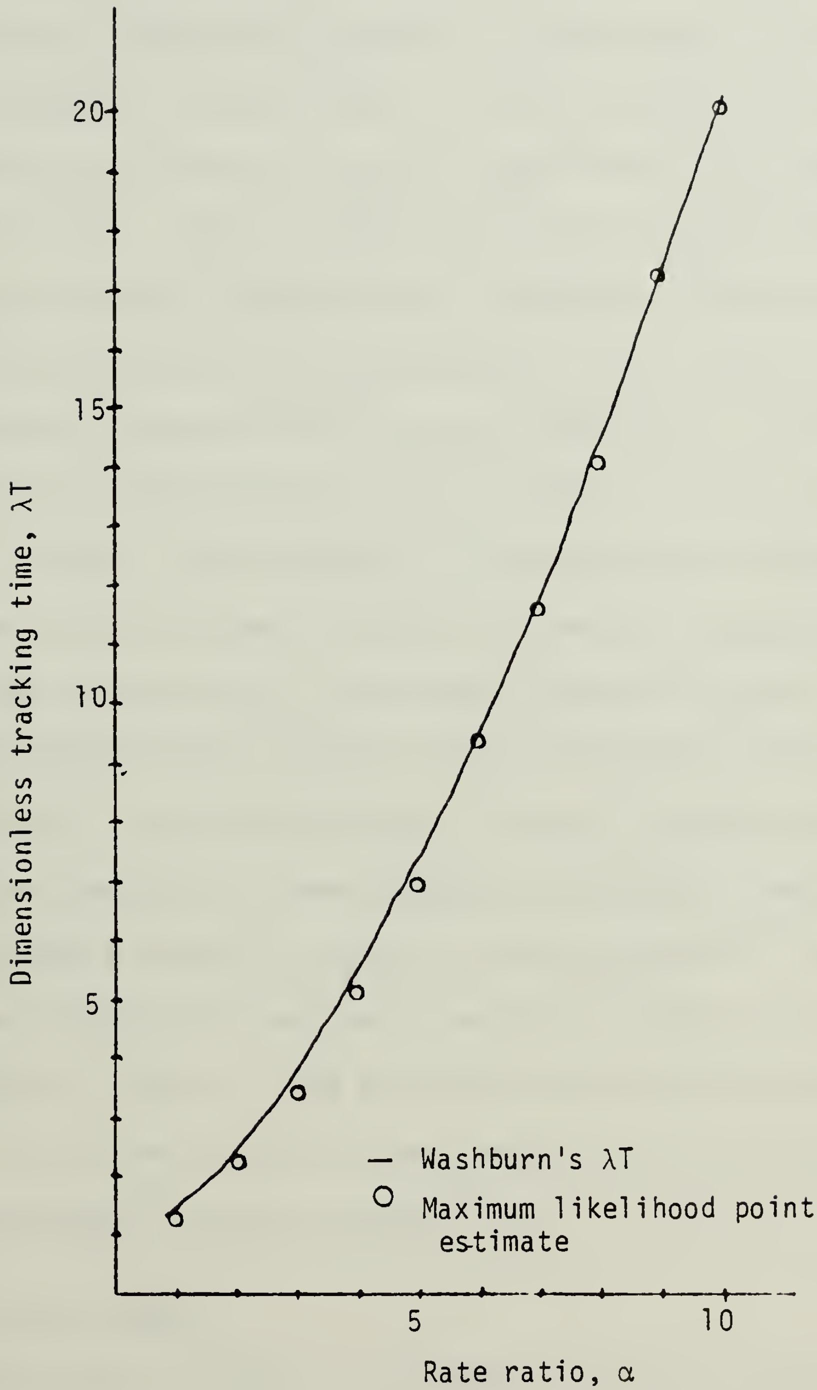

Figure 3. Dimensionless tracking time, λT, as a function of rate ratio, α.

The continuous time Fortran simulation listed at the end of this report was designed to examine the effectiveness of the maximum likelihood decision rule. The dynamic parameters of the program include the component parameters of the rate ratio, α, and the accuracy ratio, β, as well as a memory input which allows for tracker memory set between one and ten detections. The case where the tracker's maximum memory capacity is equal to one is the equivalent of Washburn's memoryless tracker. Using the maximum likelihood memoryless tracker model, it is possible to generate a graph similar to that of Washburn's. Point estimates, based on 1000 iterations at each α level are marked with a o in figure 3. Table I shows Washburn's λT, the maximum likelihood point estimates, a 95% confidence interval for the point estimates, and the sample standard deviation for the maximum likelihood results.

The results indicate that tracking time for the maximum likelihood memoryless tracker is roughly exponential in nature, with the sample mean approximately equal to the sample standard deviation. More importantly, the times are nearly as good as those of Washburn. Washburn's model was specifically designed to maximize tracking time as it is defined in his report. Using the same definition for tracking time, we are able to nearly duplicate his results with the simplest version of the maximum likelihood tracker.

B. THE TRACKER WITH MEMORY

Having established a tracking algorithm that is nearly the equal of an optimized memoryless tracker, our goal will now be to improve upon the

TABLE I. Dimensionless tracking value, λT, as a function of rate ratio, α; $\beta=0$.

α	Washburns λT	Maximum Likelihood λT	95% Confidence Interval Lower Bound	Upper Bound	Sample Standard Deviation
1	1.4	1.21	1.15	1.27	1.02
2	2.35	2.33	2.22	2.44	1.83
3	3.8	3.52	3.34	3.70	2.94
4	5.5	5.22	4.93	5.51	4.65
5	7.35	6.95	6.56	7.34	6.22
6	9.45	9.35	8.81	9.89	8.72
7	11.9	11.28	10.62	11.94	10.60
8	14.4	14.04	13.12	14.95	13.36
9	17.1	17.20	16.11	18.29	17.52
10	20.1	20.08	18.92	21.24	18.69

TABLE II. Dimensionless tracking value, λT, as a function of memory; $\alpha=4$, $\beta=0$.

Memory	λT	95% Confidence Interval on λT Lower Bound	Upper Bound	% Improvement $100\cdot\left(\frac{\lambda T-5.39}{5.39}\right)$
1	5.39	5.08	5.70	—
	Washburn's $\lambda T = 5.45$			
2	5.53	5.23	5.83	2.3
3	5.84	5.54	6.14	8.3
4	5.86	5.54	6.18	8.7
5	5.82	5.49	6.15	7.9
6	5.90	5.57	6.23	9.5
7	5.90	5.59	6.21	9.5
8	5.58	5.28	5.88	3.5
9	6.13	5.78	6.48	13.7
10	5.90	5.58	6.22	9.5

model by introducing a memory capacity. As stated in part A of this chapter, the simulation used in this report was run with as many as ten detections on file. At each detection the model evaluates every possible label combination for the detections in memory and accepts the most likely. Note that the track selected will either include the most recent detection or be null. In accordance with the tracking rules, when a non-null track is accepted all but the most recent position estimate are scrubbed from memory. The result is a type of time sequenced problem where the number of label combinations to be examined actually doubles as each new detection is added to memory. For the purpose of this report, a memory capacity of ten detections was found to sufficiently illustrate the benefit of memory. Further increases in memory capacity are made at a substantial cost in computer processing time. As will be shown later in this report, the small increase in tracking time is not likely to be worth the extra effort required for more memory. There appears to be an upper bound on tracking ability. The bound is nearly reached with a memory capacity less than ten. As the capacity increases tracking time asymptotically approaches the upper bound.

Because of the limited storage capacity, the possibility does exist that memory capacity might be exceeded before a detection, or set of detections, is accepted. In this case we will have to introduce a procedure whereby detections can be selectively purged from memory. The question of where to begin memory cleansing stimulates several possible approaches. Not only individual detections, but combinations

of several detections could be considered to determine which might offer the greatest chance of future acceptance. For simplicity, in this report the detections will be considered individually, and the single most unlikely detection will be scrubbed.

Looking now at a specific example, we begin to observe the effect of memory. In the case where $\alpha=4$, $\beta=0$, the incremental effects of memory are illustrated in figure 4. A curve, based on 1000 iterations at each memory level, has been fitted between the points to roughly approximate the shape of the empirical solution. Although not smooth, the curve does indicate a significant gain in tracking time as memory capacity is initially expanded to several detections. The curve then seems to flatten somewhat, indicating a diminishing return for memory. This flattening of the curve, found in every sample run, provided justification for limiting memory capacity to ten detections. In fact, if computer access were limited, a memory capacity of three or four detections might prove sufficient. Table II lists the point estimates, 95% confidence intervals, and the per cent improvement over base case, memory = 1.

Although we have reported only one case, the effects of memory were found to be universal. Various runs with α ranging from one to ten showed the maximum likelihood tracker could consistently outperform Washburn's memoryless tracker when equipped with a memory capacity as small as two or three detections. In the next chapter we will examine the method by which the maximum likelihood tracker is able to handle the problem of measurement error.

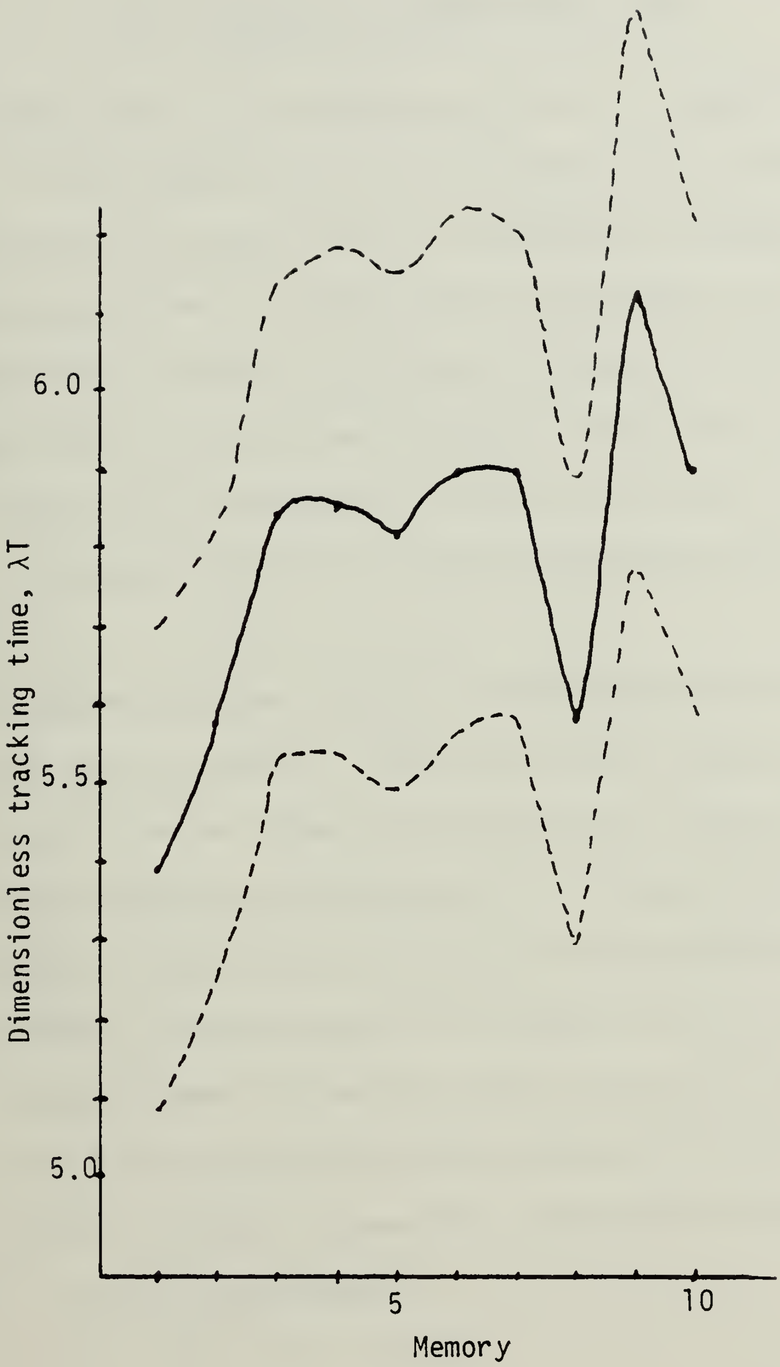

Figure 4. Dimensionless tracking time, λT, and 95% confidence intervals as a function of memory; $\alpha=4$, $\beta=0$.

IV. TRACKING WITH MEASUREMENT ERROR

Measurement error in the detection system is a common problem for trackers. In this report the error is assumed to be circular normal. This distribution is commonly used in modeling both because of its appropriateness for many existing systems and because of its mathematical tractability. In this section we will discuss the effects of measurement error on the maximum likelihood tracker and expand the model to deal with it accordingly. In addition, we will examine the results of a computer simulation in which measurement error was included.

To illustrate the added dimension that measurement error imposes, we can first look at figure 5. The probabilistic term for real target position reports can no longer be defined with the simple bivariate normal motion equation used in chapter 2. Instead, we will define a new expression which accounts for both the motion and the error between subsequent position reports. In figure 5, $R_{i,i+1}$ will be the vector distance between the target's actual position at times i and i+1. R_i will be the measurement error associated with the iTH detection. The vector distance between real target detections will be called Q_i. In this case we have

$$Q_1 = - R_1 + R_{1,2} + R_2$$

$$Q_2 = - R_2 + R_{2,3} + R_3$$

$$\vdots$$

$$Q_i = - R_i + R_{i,i+1} + R_{i+1}$$

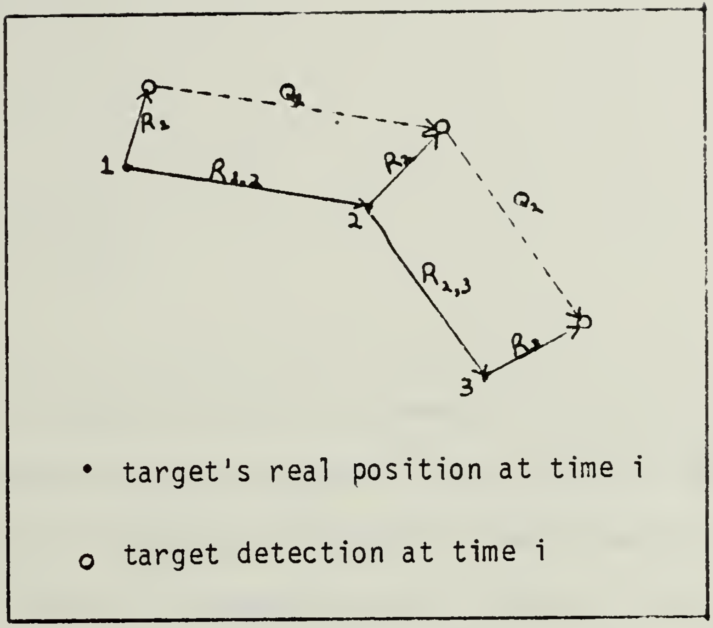

Figure 5. Sample set of target position reports and associated target positions.

Since the problem is symmetric in the X and Y dimension, for simplicity we can examine a one dimensional version of the same problem. Since the X and Y components of target motion and error are independent and parameterized identically, we will be able to combine terms to form a single solution based on the identically distributed but independent results.

In this case the one dimensional, horizontal distance between detections will be called X_i. The error associated with the iTH detection will be called E_i, and the distance between actual target position i and i+1 will be V_i. A series of measurements can be represented as follows:

$$X_1 = - E_1 + V_1 + E_2$$

$$X_2 = - E_2 + V_2 + E_3$$

$$\vdots$$

$$X_i = - E_i + V_i + E_{i+1}$$

$$\vdots$$

We know the E and V variables are all independent with means equal to zero, that the variance of E_i is equal to σ^2, and that the variance of V_i is equal to Dt_i. However, the individual measurements, X_i, are not independent. Since E_2 is common to both X_1 and X_2, $E[X_1 \; X_2] = - \sigma^2$, and more generally $E[X_i \; X_{i+1}] = - \sigma^2$. Since $E[X^2] = 2\sigma^2 + Dt_i$, the covariance matrix of $U_3 = (X_1, X_2, X_3)^t$, for instance, will be

$$\Sigma_3 = \begin{bmatrix} 2\sigma^2 + Dt_1 & -\sigma^2 & 0 \\ -\sigma^2 & 2\sigma^2 + Dt_2 & -\sigma^2 \\ 0 & -\sigma^2 & 2\sigma^2 + Dt_2 \end{bmatrix}$$

This matrix retains its form, expanding only along the diagonals, as the dimension is increased. Since the vector $U_i = (X_1, X_2, \ldots, X_i)^t$ is multivariate normal, the probability density function of U_i will be

$$f(U_i) = (\sqrt{2\pi}^{\,i} \sqrt{|\Sigma_i|}^{\,-1}) \exp[- \tfrac{1}{2} J_i^{\,U}]$$

where $J_i^U = U_i^t \Sigma^{-1} U_i$ and $|\ |$ denotes determinant [4]. By symmetry, the same probability density function holds for the vector of incremental measurements in the vertical direction, $W_i = (Y_1, Y_2, \ldots, Y_i)^t$. Since U_i and W_i are independent, the joint probability density function is obtained by multiplication:

$$f(U_i,W_i) = ((2\pi)^i\ |\Sigma_i|)^{-1} \exp[-J_i/2]$$

where $J_i^W = W_i^t \Sigma^{-1} W_i$ and $J_i = J_i^U + J_i^W$. This then will replace the term used in the decision rule developed in chapter 2 . In final form, the new expression will be:

$$L_i = \frac{1}{2} J_i + \ln[(2\pi m/\lambda)^i\ |\Sigma_i|]$$

Note that this is the equivalent of (12) from chapter (2) when σ equals zero.

Although the decision rule looks relatively simple when expressed in matrix notation, computer implementation of the rule creates another problem. To avoid the burden of recomputing Σ^{-1} and $|\Sigma|$ for each of the tracks examined, it is necessary to develop an iterative procedure by which a summary term can be identified for each of the tracks examined. Using a type of Kalman filter, developed mathematically in Appendix A, it is possible to obtain a position estimate, track value (L_i), and gain matrix for each of the tracks examined. If none of the tracks are subsequently accepted, the terms can be saved and used when evaluating the next detection that occurs. The procedure

is completely iterative and reduces to the procedure developed in chapter (2) when β equals zero. A flow chart for the algorithm is outlined below with $S^2 = Dt_i$ and $d = |\Sigma|$. See Appendix A for the definition of v, x, and y.

$$J = 0$$

$$v = \infty$$

$$x = 0$$

$$y = 0$$

$$d = 1$$

$$\rightarrow \text{Observe } X, Y, \text{ and } S^2$$

$$v = 2\sigma^2 + S^2 - \sigma^4/v$$

$$d = d \cdot v$$

$$J = J + [(x-X)^2 + (y-Y)^2]/v$$

$$x = \sigma^2(x-X)/v$$

$$y = \sigma^2(y-Y)/v$$

To test the effectiveness of the revised decision rule, we can use the Fortran simulation listed at the end of this report. For a sample case where α=4 and β=1, the sample results from 1000 iterations at each memory level are graphed in figure 6. Point estimates, 95% confidence intervals, sample standard deviation, and the per cent improvement over base case (memory = 1) are listed in Table III. The results are as we might intuitively expect. Memory capacity again appears to benefit the tracker, but the gain appears to level off as memory is

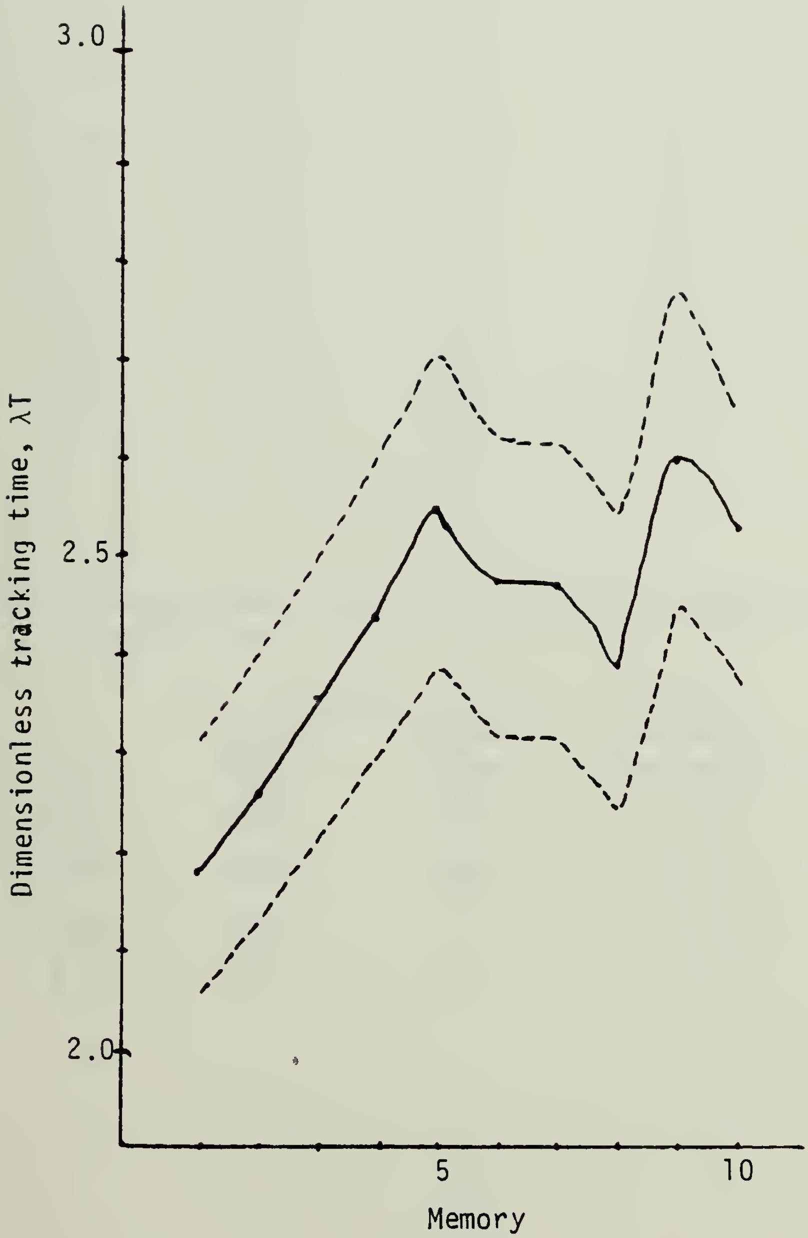

Figure 6. Dimensionless tracking time, λT, and 95% confidence intervals as a function of memory; $\alpha=4$, $\beta=1$.

TABLE III. Dimensionless tracking time, λT, as a function of memory; $\alpha=4$, $\beta=1$.

Memory	λt	95% Confidence Interval Lower Bound	Upper Bound	Sample Standard Deviation	% Improvement
1	2.18	2.05	2.31	2.04	—
2	2.26	2.13	2.39	2.13	3.7
3	2.36	2.22	2.50	2.24	8.3
4	2.44	2.29	2.59	2.37	11.9
5	2.55	2.39	2.71	2.55	17.0
6	2.47	2.32	2.62	2.39	13.3
7	2.47	2.32	2.62	2.35	13.3
8	2.39	2.24	2.54	2.34	9.6
9	2.60	2.45	2.75	2.37	19.3
10	2.51	2.37	2.65	2.30	15.1

TABLE IV. Dimensionless tracking time, λT, as a function of memory; $\alpha=8$; $\beta=1$.

Memory	λT	95% Confidence Interval Lower Bound	Upper Bound	Sample Standard Deviation	% Improvement
1	6.17	6.05	6.29	6.04	—
2	6.53	6.40	6.66	6.46	5.8
3	6.62	6.49	6.75	6.55	7.3
4	6.58	6.45	6.71	6.39	6.6

increased beyond four or five detections. Compared to the case where $\beta=0$, the per cent improvement in tracking time appears to be better although the upper bound on tracking time is much less.

Due to the large sample standard deviation, the 95% confidence intervals are also large. Consequently, even with 1000 iterations, the point estimates do not produce the smooth curve we might have hoped to find for figure 6. To more closely approximate the nature of the curve, we will examine the results of 10,000 iterations of a test case where $\alpha=8$ and $\beta=1$. Memory will range from (1) to (4) detections. The results are graphed in figure 7, and listed in table IV. As expected, the results begin to outline the shape of a smooth curve, rising steeply at first, then tapering off to some constant value.

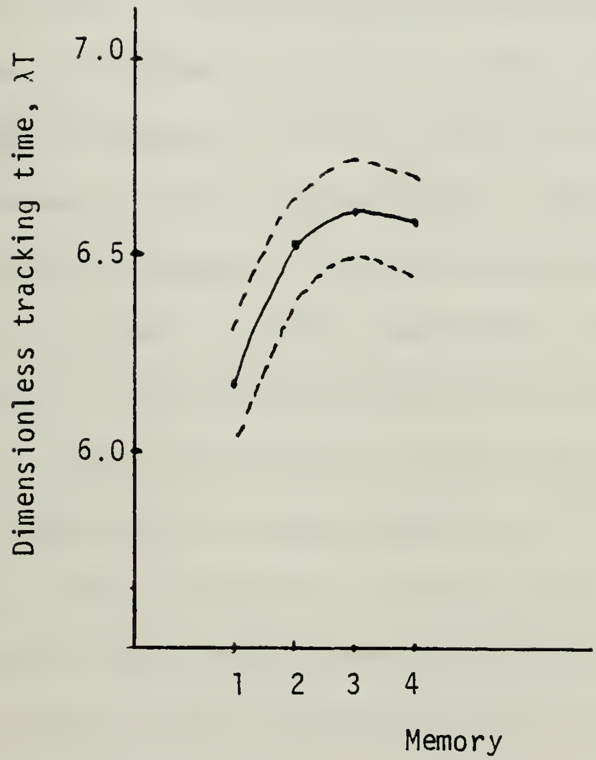

Figure 7. Dimensionless tracking time, λT, and 95% confidence intervals as a function of memory, $\alpha=8$, $\beta=1$.

V. SUMMARY

First, looking at the case where there is no measurement error ($\beta=0$), we have seen that the maximum likelihood model presented here is actually a special case of the maximum likelihood model developed by Sittler [2]. In this model, when we examine a set of detections, a dimensionless "distance" is assigned to every possible combination of real detections. The distance is based on a maximum likelihood solution, and the combination of detections with the shortest total distance is accepted as the target's real track. The method is intuitively reasonable and relatively easy to implement with a computer.

When this method is compared to the optimal memoryless tracker developed by Washburn [3], we find that the maximum likelihood memoryless tracker is nearly the equivalent. When furnished with even a limited memory capacity, we find that the maximum likelihood tracker quickly surpasses the optimal memoryless tracker in ability. However, the return of increased tracking time for additional memory diminishes quickly. There appears to be an upper bound on tracking time which is nearly reached with a memory capacity of four or five detections. In all cases the distribution of tracking times was found to be roughly exponential with the mean approximately equal to the standard deviation.

In the case where measurement error was not equal to zero, we were required to first develop a type of Kalman filter to handle the process of obtaining an iterative expression for the "distance" between

detections in a set. Once developed, we saw that the case where $\beta=0$ was really a special case of this more general problem. The tracking times still appear to follow an exponential distribution, and the payoff for memory seemed to increase as β approached α.

The question "How much memory is enough?", is particularly relevant to the available computational power as well as the importance of minor improvements in tracking times. It appears as though we can approach the bound with as few as four or five detections; however some small gain in tracking time will still be realized beyond that memory level. The cost of obtaining the small increase in tracking time is a marked increase in computer processing time.

APPENDIX A

DEVELOPMENT OF KALMAN FILTER

We will consider a one dimensional system in which position measurements are taken sequentially. Target motion between measurements is assumed to be the result of normal diffusion and will be represented by V_N, where $V_N \sim N(0, S_N^2)$ and $S_N^2 = Dt_N$. Each position measurement will also have an associated measurement error, E_N, which will also have a normal distribution; i.e. $E_N \sim N(0, \sigma^2)$. The distance between successive measurements will be;

$$Z_N = E_N + V_N - E_{N-1}; \quad N = 1, 2, \ldots$$

The symbol Z_N has replaced X_N of the text (CH-4) because we will have a special use for the symbol X_N.

Let Σ_N be the covariance matrix of $U_N = [Z_1, \ldots, Z_N]^t$ and $J_N = U_N^t \Sigma_N^{-1} U_N$. We wish to find an iterative means of producing J_N.

To begin let $X_N = [E_N, E_{N-1}]^t$, so that $X_{N+1} = \phi X_N + W_N$; where $\phi = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$, $W_N = (E_{N+1}, 0)^t$, and $\text{cov}(W_N) = \begin{bmatrix} \sigma^2 & 0 \\ 0 & 0 \end{bmatrix}$.

We also have $Z_N = HX_N + V_N$; where $H = [1, -1]$. Now let the NxN matrix L_N be a whitening filter; i.e. $L_N U_N$ is white noise. Then $I_{NxN} = \text{cov}[L_N U_N] = L_N \Sigma_N L_N^t$.

Using matrix manipulations we obtain

$$L_N^{-1}(L_N^{-1})^t = \Sigma_N$$

and

$$L_N^t L_N^{-1} = \Sigma_N^{-1}$$

therefore

$$U_N^t (L_N^t L_N) U_N = U_N^t \Sigma^{-1} U_N = J_N .$$

Now let

$$L_N U_N = (Q_1, Q_2, \ldots, Q_N)^t \text{ so that}$$

$$J_N = (L_N U_N)^t (L_N U_N) = \sum_{i=1}^{N} Q_i^2 .$$

Also let $z_N = E[Z_N | Z_1, \ldots, Z_{N-1}]$ and

$$v_N = \text{var}\,[Z_N | Z_1, \ldots, Z_{N-1}] .$$

Then

$\dfrac{Z_N - z_N}{\sqrt{v_N}}$ is independent of $Z_1, \ldots, Z_{N-1}$, normal, and has unit variance; i.e.

$$Q_N = \frac{Z_N - z_N}{\sqrt{v_N}} .$$

At this point, our intention will be to use Kalman filtering to obtain an iterative expression for z_N and v_N. Since $J_N = \sum_{i=1}^{N} Q_i^2$ and Z_N can be measured, when z_N and v_N are available we have an iterative expression for J_N. The remainder of this appendix will closely follow the Kalman filtering technique used by Gelb [1]. So that our notation will conform to his, we will first define the following terms;

$$\hat{x}_N(-) = E[X_N | Z_1, \ldots, Z_{N-1}] = \begin{bmatrix} 0 \\ -z_N \end{bmatrix}$$

$$\hat{x}_N(+) = E[X_N | Z_1, \ldots, Z_N]$$

$$P_N(-) = \mathrm{cov}[X_N | Z_1, \ldots, Z_{N-1}]$$

$$P_N(+) = \mathrm{cov}[X_N | Z_1, \ldots, Z_N]$$

To begin, let $\sigma_N^2 = \mathrm{var}\,[E_{N-1} | Z_1, \ldots, Z_{N-1}]$, so that $\sigma_1^2 = \sigma^2$ and $P_N(-) = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma_N^2 \end{bmatrix}$. Then

$$v_N = H\,P_N(-)\,H^t + S_N^2 = \sigma^2 + \sigma_N^2 + S_N^2 \;.$$

The Kalman gain matrix, K_N, is given by $K_N = \begin{bmatrix} \sigma^2/v_N \\ -\sigma^2/v_N \end{bmatrix}$. Following the Kalman format, we can update the current error estimate, $\hat{x}_N(+)$, as follows;

$$\hat{x}_N(+) = \hat{x}_N(-) + K_N[Z_N - H\,\hat{x}_N(-)]$$

$$= \begin{bmatrix} 0 \\ -z_N \end{bmatrix} + \begin{bmatrix} \sigma^2/v_N \\ -\sigma^2/v_N \end{bmatrix} [Z_N - 0 - z_N]$$

$$= \begin{bmatrix} (\sigma^2/v_N)(Z_N - z_N) \\ -z_N - (\sigma^2/v_N)(Z_N - z_N) \end{bmatrix}$$

Since $\hat{x}_{N+1}(-) = \phi\hat{x}_N(+)$ and $z_{N+1} = H\,\hat{x}_{N+1}(-)$, $z_{N+1} = H\,\phi\,\hat{x}_N(+) = -(\sigma^2/v_N)(Z_N - z_N)$ which gives the desired iterative solution for z_N. We continue working toward a solution for v_N by first updating the covariance matrix;

$$P_N(+) = (I-KH)\,P_N(-) = (I-KH)\begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma_N^2 \end{bmatrix}$$

$$= \begin{bmatrix} q_N^2 & b \\ c & d \end{bmatrix}; \qquad \text{where } q_N^2 = \sigma^2 - \sigma^4/v_N = \sigma^2(1 - \sigma^2/v_N)\;.$$

Regardless of b, c, and d, we can compute $P_{N+1}(-)$ as follows;

$$P_{N+1}(-) = \phi \; P_N(+)\phi + \text{cov}[W_N]$$

$$= \begin{bmatrix} \sigma^2 & 0 \\ 0 & q_N{}^2 \end{bmatrix} \text{ so that } \sigma_{N+1}^2 = q_N{}^2 .$$

At this point we have completed the one-dimensional analysis. The computations are in the following order (an intermediate computation of $\sigma_N{}^2$ has been eliminated):

Setting $v_0 = \infty$ to initialize the problem

we have;

$$z_1 = 0$$
$$v_1 = \sigma^2 + S_1{}^2 + \sigma^2(1-\sigma^2/v_0)$$
$$J_1 = Z_1{}^2/v_1$$

$$z_2 = \sigma^2(z_1 - Z_1)/v_1$$
$$v_2 = \sigma^2 + S_2{}^2 + \sigma^2(1-\sigma^2/v_1)$$
$$J_2 = J_1 + (Z_2-z_2)^2/v_2$$

$$\vdots$$

By using a flow chart we can eliminate the variable subscripts. The procedure will be as follows;

$$v = \infty$$

$$z = 0$$

$$J = 0$$

$\longrightarrow$ observe Z and S^2

$v = 2\sigma^2 + S^2 - \sigma^4/v$

$J = J + (z - Z)^2/v$

$z = \sigma^2 \, (z - Z)/v$

When we examine a two dimensional, symmetric version of the same problem, as considered in the text, we are actually looking at two independent but identical problems. In this case we solve for $J_N^U = U_N^t \, \Sigma_N^{-1} \, U_N$, and $J_N^V = V_N^t \, \Sigma_N^{-1} \, V_N$ using the procedure developed in this appendix. Because of the symmetric parameters, we can combine terms to form a single value J. In a flow diagram we can compute J as follows; $J = J + [(x-X)^2 + (y-Y)^2]/v$ where J is initially set equal to zero. This term can then be used in a bivariate normal probability distribution used to represent successive real detection positions. The entire flow chart for the two dimensional case is shown in chapter (4) of the text.

Using this formulation, we can also find a way to iteratively calculate the value of $|\Sigma_i|$. As previously defined $Q_N = L_N U_N$ and $\mathrm{cov}[Q] = E[Q_N \, Q_N^t] = L_N \, \Sigma_N \, L_N^t = I_{N \times N}$, so that $\mathrm{cov}[U_N] = \Sigma_N = L_N^{-1} (L_N^t)^{-1}$. Since L_N will be a lower rectangular matrix with the i^{th} diagonal element equal to $1/\sqrt{v_i}$, we have;

$$|L_N| = 1\Big/\sqrt{\prod_{i=1}^{N} v_i} = |L_N^t|$$

and therefore;

$$|\Sigma_N| = \frac{1}{|L_N|} \, \frac{1}{|L_N^t|} = \prod_{i=1}^{N} v_i \text{ , since}$$

$|AB| = |A|\ |B|$ and $|A^{-1}| = |A|^{-1}$. This then will be the iterative solution for $|\Sigma_N|$, as is also shown in the flow diagram of chapter 4.

COMPUTER PROGRAM

```
C                MAXIMUM LIKELIHOOD TRACKER
C
      COMMON DIFU,ETA,YLMDA,SIG,NPTS,TR,MRPT,IN,TM,K1
      COMMON X,Y,T,XLAST,YLAST,TLAST,ANS,SLEG,MSEL,MNPTS,PATH,NAR,LIST
      COMMON QJ,V,DET,TI(300),MLEG(30)
      COMMON XDET,YDET,LABEL
      DIMENSION XDET(1100),YDET(1100)
      DIMENSION X(1100),Y(1100),T(1100),XLAST(1100),YLAST(1100)
      DIMENSION TLAST(1100),PATH(1100),NAR(1100),LIST(6000),SLEG(30)
      DIMENSION MRPT(15)
       DIMENSION QJ(1100),V(1100),DET(1100)
      CALL OVFLOW
      WRITE(6,11)
11    FORMAT('0','ALFA',2X,'BETA',2X,'AVG T',4X,'STD DEV    DIFU',5X,'ET
     1A',6X,'YLMDA',4X,'SIG',6X,'ITR',6X,'MEMORY')
      ISEED1=245813
          ISEED2 =94082
      ISEED3=16259
      ISEED4=71302
      DO 610 MNPTS=1,4
      READ(5,1)DIFU,ETA,YLMDA,SIG,ITR
C
1      FORMAT(4F10.5,I5)
      ALFA=YLMDA/SQRT(2*3.141592654*ETA*DIFU)
      BETA=SIG/SQRT(DIFU/YLMDA)
         TSTD=0.
         TTEMP=0.
      DO 500 LM=1,ITR
C
C    SET INITIAL VALUES
      PATH(1100)=0.
      XLAST(1100)=0.
      YLAST(1100)=0.
      TLAST(1100)=0.
      X(1100)=0.
      Y(1100)=0.
      T(1100)=0.
      NAR(1100)=0
      CALL NORMAL(ISEED1,Q1,1)
      CALL NORMAL(ISEED1,Q2,1)
      XOLD=Q1*SIG
      YOLD=Q2*SIG
      TOLD=0.
      TFOLD=0.
      NPTS=1
      IN=0
      JREJ=0
      NF=0
```

```
      K1=0
      TM=0.
      V(1100)=10**8
      QJ(1100)=0.
      DET(1100)=1.
      XDET(1100)=0.
      YDET(1100)=0.
       LABEL=0
C
C      GENERATE TIME/POSIT OF NEXT REAL DETECTION
20    CALL EXPON(ISEED1,Z,1)
      TR=Z/YLMDA+TOLD
      CALL NORMAL(ISEED2,Q1,1)
      CALL NORMAL(ISEED2,Q2,1)
      CALL NORMAL(ISEED2,Q3,1)
      CALL NORMAL(ISEED2,Q4,1)
      REALP=SQRT(DIFU*TR)
      XR=(Q1*REALP)+(Q3*SIG)+XOLD
       YR=Q2*REALP+Q4*SIG+YOLD
C       ACTUAL POSIT DIFFERS BY SIG FROM DETECTION POSIT
      XPOS=REALP*Q1+XOLD
      YPOS=REALP*Q2+YOLD
C
C      IS A FALSE DETECTION SKED
      IF(NF .EQ. 1) GO TO 40
C
C      GENERATE TIME OF NEXT FALSE DETECTION
25     CALL RANGE(RF)
      CALL EXPON(ISEED3,Z1,1)
      TF=Z1/(3.141592654*RF*RF*ETA) +TFOLD
      NF=1
C
C      GENERATE POSIT OF NEXT FALSE DETECTION
C
30    CALL RANDOM(ISEED4,Z3,1)
      CALL RANDOM(ISEED4,Z4,1)
      XF=2*RF*Z3-RF
      YF=2*Z4*RF-RF
      TEM=SQRT(XF*XF+YF*YF)
      IF(TEM .GT. RF) GO TO 30
C
C      TAKE NEXT EVENT , SET VALUES FOR LSTSQR
40    LABEL=LABEL +1
      IF (TR .LT. TF) GO TO 50
      TI(LABEL)=TF
      NTYPE=1
      X(NPTS)=XF
      Y(NPTS)=YF
```

```
      T(NPTS)=TF
      TFOLD=TF
      GO TO 55
50    NTYPE=0
      TI(LABEL)=TR
      X(NPTS)=XR
      Y(NPTS)=YR
      T(NPTS)=TR
      TOLD=TR
      IN=IN+1
      MRPT(IN)=LABEL
C
55    CALL LSTSQR
C
      IF (MSEL .EQ. 1100) GO TO 200
C
      IF(MSEL .EQ. 1099) GO TO 498
C    STOP TRACKING TIME AT TIME OF 1ST MISSED DETECTION
C   CHECK ALL PTS IN PATH FOR ERROR
C
60    M2=0
C   CALL PATH PTS FOR PATH(MSEL)
      IF(MSEL .EQ. 1 ) GO TO 71
      L5=MSEL-1
      DO 70 M3=1,L5
         M2=M2+NAR(M3)
70    CONTINUE
71    CONTINUE
      M4=1
      L6=M2+1
      L7=M2+NAR(MSEL)
      IF(IN .EQ. 0) GO TO 85
      DO 80 M5=L6,L7
       IF(M4 .GT. IN) GO TO 90
         IF(MRPT(M4) .NE. LIST(M5)) GO TO 90
         M4=M4+1
80    CONTINUE
      GO TO 100
C
C   PATH NOT CORRECT STOP TRACKING
85    IF(MSEL .EQ. 1) GO TO 88
      LN=0
      KN=MSEL-1
      DO 87 I=1,KN
         LN=LN+NAR(I)
87    CONTINUE
      TM=TM+TI(LIST(LN+1))
      GO TO 93
```

```
88    TM=TM+TI(LIST(1))
      GO TO 93
C
90    TM=TM+TI(MRPT(M4))
      K1=K1+M4-1
93    GO TO 498
C
100   TM=TM+TI(MRPT(IN))
       LABEL=0
      NPTS=1
      K1=K1+IN
      NF=0
      XOLD=-SIG*Q3
      YOLD=-SIG*Q4
      IN=0
      JREJ=0
      TOLD=0.
      TFOLD=0.
      XLAST(1100)=XLAST(MSEL)
      YLAST(1100)=YLAST(MSEL)
      V(1100)=V(MSEL)
      GO TO 20
C
200   IF(NTYPE .EQ. 0) GO TO 210
       NPTS=NPTS+1
      NF=0
      GO TO 25
C
C      OFFSET STARTING POINT , MISSED A REAL DETECTION
210   XOLD=XPOS
      YOLD=YPOS
      NPTS=NPTS+1
      JREJ=JREJ+1
C
C     LIMIT NUMBER OF REAL DETECTIONS MISSED
C
      IF(JREJ .LT. 11) GO TO 220
       TM=TM+TI(MRPT(1))
      GO TO 498
220   GO TO 20
C
C
498   TTEMP=TTEMP+TM
      TSTD=TSTD+TM*TM
500   CONTINUE
C
      AVGT=TTEMP/FLOAT(ITR)
      STD=SQRT(TSTD/FLOAT(ITR)-AVGT*AVGT)
```

```
      WRITE(6,508)ALFA,BETA,AVGT,STD,DIFU,ETA,YLMDA,SIG,ITR,MNPTS
508   FORMAT('0',F4.1,2X,F4.1,2X,6(F7.3,2X),2(I5,2X))
C
610   CONTINUE
      STOP
      END
```

```
      SUBROUTINE LSTSQR
C
      COMMON DIFU,ETA,YLMDA,SIG,NPTS,TR,MRPT,IN,TM,K1
      COMMON X,Y,T,XLAST,YLAST,TLAST,ANS,SLEG,MSEL,MNPTS,PATH,NAR,LIST
      COMMON QJ,V,DET,TI(300),MLEG(30)
      COMMON XDET,YDET,LABEL
      DIMENSION XDET(1100),YDET(1100)
       DIMENSION QJ(1100),V(1100),DET(1100)
      DIMENSION X(1100),Y(1100),T(1100),XLAST(1100),YLAST(1100)
      DIMENSION TLAST(1100),PATH(1100),NAR(1100),LIST(6000),SLEG(30)
      DIMENSION MRPT(15)
C
C      COMPUTE NO. OF NEW PATHS
      IF(NPTS .EQ. 1) GO TO 10
      NEWP=2*NEWP
      GO TO 11
10    NPATH=0
      NEWP=1
C
C     LABEL PATHS THAT NEED CHECKED , LOW-NUP
11    LOW=NPATH+1
      NUP=NPATH+NEWP
C
C       COMPUTE NEW PATH VALUES, STORE LAST X,Y, AND T
      DO 20 I=LOW,NJP
         M=I-LOW
      IF(M .EQ. 0) M=1100
         CALL VALUE (M,I)
            PATH(I)=ANS
20    CONTINUE
C
C    STORE POINTS IN EACH PATH
       IF(NPTS .EQ. 1)MPTS=0
C    FIRST COMPUTE NO. OF PTS IN EACH NEW PATH
       NAR(LOW)=1
       IF(LOW .EQ. 1) GO TO 31
       K6=LOW+1
       DO 30 K=K6,NUP
          M=K-LOW
          IF(M .EQ. 0) M=1100
          NAR(K)=NAR(M)+1
30    CONTINUE
31    NALL=2*MPTS+NEWP
C
      K=LOW
      N1=0
      N2=1
       N3=MPTS+1
```

```
      DO 40 L=N3,NALL
         N1=N1+1
         IF(N1 .EQ. NAR(K)) GO TO 35
         LIST(L)=LIST(N2)
         N2=N2+1
         GO TO 40
35       LIST(L)=LABEL
         N1=0
         K=K+1
40    CONTINUE
C
C     SAVING SINGLE LEG PATH VALUES
      SLEG(NPTS)=PATH(LOW)
      MLEG(NPTS)=LABEL
C
C    UPDATE COUNT OF TOTAL PTS/PATHS STORED
C
      MPTS=NALL
       NPATH=NPATH+NEWP
C   PICK SMALLEST PATH
C
      MSEL=1100
      DO 50 I= LOW,NUP
         IF(PATH(I) .GT. PATH(MSEL)) GO TO 50
         MSEL=I
50    CONTINUE
C   CHECK IF PATH ACCEPTED
C
      IF(MSEL .NE. 1100) GO TO 100
C
      IF(NPTS .LT. MNPTS) GO TO 100
C
C     OTHERWISE MUST PURGE A POINT REMOVE ALL PATHS W PT INCLUDED
      CALL PURGE(NEWP,NPATH,MPTS)
100   RETURN
      END
```

```
      SUBROUTINE VALUE (M,I)
C
      COMMON DIFU,ETA,YLMDA,SIG,NPTS,TR,MRPT,IN,TM,K1
      COMMON X,Y,T,XLAST,YLAST,TLAST,ANS,SLEG,MSEL,MNPTS,PATH,NAR,LIST
      COMMON QJ,V,DET,TI(300),MLEG(30)
      COMMON XDET,YDET,LABEL
      DIMENSION XDET(1100),YDET(1100)
       DIMENSION QJ(1100),V(1100),DET(1100)
      DIMENSION X(1100),Y(1100),T(1100),XLAST(1100),YLAST(1100)
      DIMENSION TLAST(1100),PATH(1100),NAR(1100),LIST(6000),SLEG(30)
      DIMENSION MRPT(15)
C
      XDET(I)=X(NPTS)
      YDET(I)=Y(NPTS)
      XNM=XDET(M)+XLAST(M)
      YNM=YDET(M)+YLAST(M)
      CT=T(NPTS)-TLAST(M)
       V(I)=2.*SIG*SIG+DIFU*CT-SIG**4/V(M)
      DET(I)=V(I)*DET(M)*6.283185308*ETA/YLMDA
      QJ(I)=QJ(M)+((X(NPTS)-  XNM    )**2+(Y(NPTS)-  YNM    )**2)/V(I)
      ANS=.5*QJ(I)+ALOG(DET(I))
      XLAST(I)=SIG*SIG*(  XNM - X(NPTS)  )/V(I)
      YLAST(I)=SIG*SIG*( YNM -Y(NPTS)    )/V(I)
      TLAST(I)=T(NPTS)
      RETURN
      END
```

```
      SUBROUTINE PURGE(NEWP,NPATH,MPTS)
C
      COMMON DIFU,ETA,YLMDA,SIG,NPTS,TR,MRPT,IN,TM,K1
      COMMON X,Y,T,XLAST,YLAST,TLAST,ANS,SLEG,MSEL,MNPTS,PATH,NAR,LIST
      COMMON QJ,V,DET,TI(300),MLEG(30)
      COMMON XDET,YDET,LABEL
      DIMENSION XDET(1100),YDET(1100)
      DIMENSION X(1100),Y(1100),T(1100),XLAST(1100),YLAST(1100)
      DIMENSION TLAST(1100),PATH(1100),NAR(1100),LIST(6000),SLEG(30)
      DIMENSION MRPT(15)
       DIMENSION QJ(1100),V(1100),DET(1100)
C
C     IDENTIFY POINT TO BE PURGED , (N)
      N=1
      DO 10 I=1,NPTS
      IF(SLEG(I) .GT. SLEG(N)) N=I
10    CONTINUE
      N8=MLEG(N)
      IF (IN .EQ. 0) GO TO 16
14    DO 15 I=1,IN
         IF(MRPT(I) .EQ. N8) GO TO 200
15    CONTINUE
16    IF(MNPTS .EQ. 1) GO TO 80
         MNUM=0
      LNUM=0
      MTEMP=0
      NTEMP=0
      K=0
      M1=0
      DO 60 I=1,NPATH
         JKL=NAR(I)
         DO 40 J=1,JKL
            IF(LIST(K+J) .EQ. N8) GO TO 50
40       CONTINUE
         M1=M1+1
         NAR(M1)=NAR(I)
         XLAST(M1)= XLAST(I)
         YLAST(M1)= YLAST(I)
         TLAST(M1)= TLAST(I)
         PATH(M1)=PATH(I)
         V(M1)=V(I)
         DET(M1)=DET(I)
         QJ(M1)=QJ(I)
         XDET(M1)=XDET(I)
         YDET(M1)=YDET(I)
         KNUM=LNUM+NAR(I)
      JNUM=LNUM+1
      DO 45 M9=JNUM,KNUM
```

```
            MNUM=MNJM+1
            LIST(M9)=LIST(MNUM)
45        CONTINUE
            LNUM=LNJM+NAR(I)
         GO TO 55
50      NTEMP=NTEMP+NAR(I)
         MTEMP=MTEMP+1
         MNUM=MNUM+NAR(I)
55       K=K+NAR(I)
60    CONTINUE
         IF(N .EQ. NPTS) GO TO 65
         J7=NPTS-1
         DO 43 J6=N,J7
            SLEG(J6)=SLEG(J6+1)
            MLEG(J6)=MLEG(J6+1)
43       CONTINUE
65    MPTS=MPTS-NTEMP
       NPATH=NPATH-MTEMP
      NEWP=NEWP/2
      GO TO 90
80    MPTS=1
      NEWP=1
      NPATH=1
90    NPTS=NPTS-1
      RETURN
C      PURGED A REAL POINT  STOP TRACKING
200   TM=TM+TI(MRPT(1))
      MSEL=1099
      RETURN
      END
```

```
      SUBROUTINE RANGE(RF)
C
      COMMON DIFU,ETA,YLMDA,SIG,NPTS,TR,MRPT,IN,TM,K1
      COMMON X,Y,T,XLAST,YLAST,TLAST,ANS,SLEG,MSEL,MNPTS,PATH,NAR,LIST
      COMMON QJ,V,DET,TI(300),MLEG(30)
      COMMON XDET,YDET,LABEL
      DIMENSION X(1100),Y(1100),T(1100),XLAST(1100),YLAST(1100)
      DIMENSION TLAST(1100),PATH(1100),NAR(1100),LIST(6000),SLEG(30)
      DIMENSION MRPT(15)
       DIMENSION QJ(1100),V(1100),DET(1100)
C
      RF=2.5*(SQRT(3.*DIFU/YLMDA)+SIG)
      RETURN
      END
```

LIST OF REFERENCES

1. Gelb, A., Applied Optimal Estimation, p. 105-119, M.I.T. Press, 1975.

2. Sittler, R. W., "An Optimal Data Association Problem in Surveillance Theory," IEEE Transactions on Military Electronics, Systems Science Issue, April, 1964.

3. Naval Postgraduate School Report NPS55-80-014, Analysis of the Memoryless Tracker, by A. R. Washburn, April, 1980.

4. Zehna, P. W., Probability Distributions and Statistics, p. 268, Allyn and Bacin Inc., 1970.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Director Strategic Systems Project Office Attn: Code sp 2021 1931 Jefferson Davis Highway Arlington, Virginia 20376	1
4.	ORI, Inc. 1400 Spring Street Silver Spring, Maryland 20910	1
5.	Assoc. Professor Alan Washburn, Code 55Ws Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
6.	Professor Nagel Forrest, Code 55Fo Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
7.	Lt Robert D. Conrad, USN Naval Education and Training Command Newport, Rhode Island 02840	2
8.	Assoc. Professor Patricia A. Jacobs, Code 55Jc Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
9.	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1

Thesis
C743
c.1

192349

Conrad
A maximum likelihood tracker.